UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN REPORT NO. 3430	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY		5. TYPE OF REPORT & PERIOD COVERED 1 June 76 - 31 August 76 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) M. Beeler, J. Burchfiel, R. S. Nickerson, R. Viswanathan, J. Makhoul, A. W. F. Huggins		8. CONTRACT OR GRANT NUMBER(s) MDA903-75-C-0180
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street, Cambridge, MA 02138		12. REPORT DATE September 1976 13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

packet radio, computer communications, PDP-11 TCP, station gateway, ELF, BCPL, cross-radio debugging, speech compression, vocoder, linear prediction, variable data rate compression, speech-quality evaluation.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This document describes progress on (1) the development of a packet radio network, (2) speech compression and evaluation. Activities reported under (1) include work on Station Software and Internetworking Research and Development; under (2) development of variable frame rate transmission schemes for pitch and gain; perceptual modeling of speech to transmit LPC parameters at a minimum average frame rate, but still maintaining good speech quality; generation of stimulus tapes and collection of rating data in a new subjective quality evaluation study.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS

I. INTRODUCTION

The packet radio project relies heavily on station software for a variety of control, coordination and monitoring functions. The role of BBN in developing this software is to specify, design, implement and deliver programs which implement these functions.

At the close of the previous quarter we were beginning to run the station software on SRI equipment. The SRI testbed is significantly more complex than the development testbed at BBN, as described in section III. The early part of this quarter was used in extensive checkout in that more demanding environment. Because of the cross network debugging tools we have developed as part of the packet radio project, we were able to perform these checkout activities on SRI equipment, 3000 miles away, while continuing our other daily tasks at BBN.

When checkout was complete and documentation had been prepared and delivered to SRI, we scheduled a week of delivery activities. While we prepared to travel to SRI, they had the opportunity to peruse the documentation. The week of delivery work went smoothly. This transfer of software and attendant documentation and familiarization activities is the major accomplishment of this quarter, and stands out as a major milestone in the packet radio project as a whole.

In addition to software preparation, BBN continues to contribute to progress of the packet radio project in areas of design study, support, and hardware deployment. This is detailed in following sections dealing with publications and with hardware.

II. MEETINGS AND PUBLICATIONS

During this quarter BBN personnel were engaged in several activities at the National Computer Conference. Besides attending the conference, we presented a paper on gateway design entitled, "Gateway Design for Computer Network Interconnection;" we participated in a panel discussion on internetworking issues; and we attended the Packet Radio Working Group meeting held at the NCC.

An important part of the documentation delivered to SRI this quarter is the Packet Radio Station Operator's Manual. This manual describes in concise, yet thorough terms how to load and start the various modules, or processes, within the station; how to communicate operational parameters to those which require them; and how to use each module in a functioning packet radio network. As an companion to this manual we prepared and delivered a commented typescript of a naive user's actual experience in bringing up the packet radio station.

A number of Packet Radio Temporary notes were issued by BBN this quarter. Their subjects exemplify both the documentation activities related to the software delivery and the ongoing research and design in which BBN continues to occupy a leading position. The PRTNs issued during this quarter are:

* PRTN 174, revision 1 - "Packet Radio Network Station Labeling Process," which includes the update described in section III.

* PRTN 180 - "Cross Radio Debugger," which documents the design of the command language and user interface. This supercedes a previous BBN PRTN, number 145, which specified a preliminary design of the cross radio debugging facility.

* PRTN 182 - "Packet Radio Information Service," which specifies the information service to be implemented in the station, and discusses operational considerations which lead to the conclusion that a second implementation, when the packet radio network is larger, will include an information service host. The design of such a service host is discussed.

* PRTN 183 - "Neighbor Table Measurements for Control of the PRN," in which BBN takes the initiative to motivate and propose a specific facility for measuring the actual network performance and relaying these measurements to the station, where global decisions may be made based on these measurements.

* PRTN 184 - "Preliminary Functional Specification of the Station Measurement Process," in which BBN again forges new concepts of experimental runs, experiment parameter control, and linguistics for naming and manipulating groups of devices.

* PRTN 185 - "Report of Station Software Delivery," issued jointly by SRI at the conclusion of the week of familiarization and transfer activities. This PRTN documents the peak accomplishment of this quarter, the successful delivery of the first complete station software.

* PRTN 186 - "Proposed Augmentation of the TCP Windowing Strategy," in which BBN addresses problems encountered in actual use of the TCP and proposes a specific remedy. The discussion of the strategy involved motivates and explains the new proposal, which is also defined in this PRTN.

* PRTN 191 - "Terminal-On-Packet Proposal," containing a clean solution to a problem which PRWG contractors have been aware of for several months but whose resolution had, until now, escaped everyone. The mechanism proposed obsoletes the stopgap measure of manual entry of terminal to PR correspondence by the station operator.

In addition to the above meetings and publications, during this quarter we consulted with Collins Radio on the design of Channel Access Protocol version 3 (CAP3), which they are implementing in the PRs. By reviewing their plans and discussing them with Collins personnel we hope to identify any difficulties or deficiencies before they arise in use, and thus speed the implementation and deployment of the packet radio network.

III. STATION SOFTWARE

This quarter's activity centered around delivering a set of operational station software to SRI. Though the initial versions of the connection, control, gateway, and debug processes were essentially completed during the last quarter, some improvements were made prior to delivery. The software was then exercised extensively in the SRI testbed and delivered to SRI. Since the delivery we have continued to support and develop the station.

A. Enhancements

The cross-radio debugger program was reorganized so that its handling of network events (e.g. lack of an acknowledgement) was better coordinated with its interaction with the user. The previous asynchronous approach to handling events was found to cause confusion in many cases.

The control process was enhanced to notice (based on the absence of ROPs) losses of connectivity, and relabel any PRs whose routes included the broken link. This gave it the ability to reroute around failed repeaters and track mobile terminals. The latest functional design was documented in Revision 1 of PRTN 174, "Packet Radio Network Station Labeling Process."

B. Testing and Delivery

Working from BBN, we spent more than a week running our software in the SRI testbed. SRI worked with us during these tests both to monitor equipment at their end and to gain some familiarity with the operation of the station software. The setup consisted of the SRI station and four PRs cabled together in the lab. Testing at SRI in addition to BBN was essential prior to delivery for several reasons:

-The SRI station had less memory than the one at BBN. We had to determine what modules would fit in and run together.

-With only two PRs at BBN, we had had no opportunity to try out labeling of multiple PRs at multiple levels and with various connectivities.

-The more complicated environment made it likely that subtle bugs could be found, which would never turn up unless many things were happening at once.

The connection, control, and debug processes were heavily exercised, and several bugs were fixed and improvements made as a result.

The software, along with documentation, was then transferred to SRI over the Arpanet, and SRI began learning to use it. We then spent a week at SRI, to formally complete the delivery. During

the week we tested and demonstrated the connection, control, debug, and gateway processes. These demonstrations corresponded to Local Area Demonstrations 4 (forwarding), 5 (remote debugging), 6 (labeling), and 7 (gateway). All software worked well; the few bugs which turned up were minor and were fixed on the spot. The detailed demonstration scenario and results of the week's activity are documented in PRTN 185, "Report of Station Software Delivery," which we issued jointly with SRI.

C. Continuing Support and Development

Two changes have been made to the control process:

1. A problem which prevented it from labeling the network if it was already labeled, but with a different station ID, was fixed. This had caused difficulties for SRI when they started the station after they had been working with the PRs manually.

2. The control process can now handle Terminal-On-Packets (TOPs) as described in PRTN 191, "Terminal-On-Packet Proposal." This enables it to automatically determine the correspondence between the IDs of end devices and those of their attached PRs. This mapping, which is necessary for forwarding traffic to the end devices, previously had to be manually entered. As of yet, there are no devices which emit TOPs.

Several days were devoted to working with SRI personnel, who were having problems using the station gateway. The difficulty was finally traced to a problem in the station PR, rather than in the station software.

IV. INTERNETWORKING RESEARCH AND DEVELOPMENT

A. Upgrading Debuggers and Bootstraps

The cross-net debuggers and bootstraps have been modified to work in an internet environment. The immediate need for this change was the transfer of our packet radio PDP-11s to the RCC net from the ARPA net. The cross-net protocol itself was not modified, but was placed within internet packets with internet headers. The bootstraps had to be pared down in order to continue to fit within the maximum size imposed by DEC's ROM loaders for the PDP-11. This required the elimination of some of the access checking in the bootstrap. Potentially, this could permit interference from simultaneous loading by multiple persons, but, in practice, there is no problem.

Prior to making this change, we prepared a revised specification of the internet packet format. This revision reordered the packet fields in order to bring the address information which all internet packets must have to the beginning of the packet. This permits protocols such as the cross-net debugger protocol to avoid having fields which are pertinent to the TCP protocol occupying space in the packet. At the same time the header space allocated for host addresses was expanded to permit all potential ARPA net hosts to be addressed.

All other programs which use internet packets have been updated to use this new format. These programs included the PDP-10 TCP, the PDP-11 TCP, and the gateways. Other implementations outside BBN have similarly been updated.

An internet packet reflector program was written to assist in debugging TCPs and similar programs. This program receives packets addressed to it, interchanges the source and destination address fields and sends them back to the original sender. While doing so, it may introduce a short or long random delay, it may duplicate the packet, or drop the packet. It also records the packets arriving and leaving to provide a record of packet flow. The use of the program has uncovered several subtle timing bugs in our TCP implementations which otherwise might have gone unnoticed for a long time.

The ELF XNCP module was modified in order support multiple network interfaces. This modification is necessary to avoid duplicating code in those cases where a gateway between two ARPA type networks is desired. The ASNSQ EMT was modified to permit the specification of a network index which selects the network to which the special queue is assigned. Another modification was to provide a mechanism whereby the cross-net debugger can siphon off internet debugging messages before they are delivered to the gateway. Logically, these messages should be dispatched to the debugger module by the gateway on the basis of their address and

format codes. But, when the gateway is the target of the debugger, it is difficult if not impossible for debugger packets to take this path. This feature shunts debugger packets to the debugger early.

There is a problem with debuggers in gateways in that the debugger must know its own address lest it intercept debugger packets destined to other target machines. To minimize the hassle of providing this address in each machine, the temporary expedient of assuming the address of the first debugger packet intercepted was adopted. This is not generally correct, but works for the current debugging efforts.

B. TCP Testing

Tests were conducted to verify that the three major implementations of TCP are compatible and that the servers available on TENEX are operational. It was found that TENEX did not handle certain error conditions in the way expected by the other two. This lead to a detailed consideration of the whole area of error handling and resulted in a number of proposed changes to the basic protocol itself.

The server most used on TENEX for TCP debugging is the ECHO program. Since this is one of the simplest user programs which can be written to run with the TCP, it withstands major protocol changes and provides a simple user-to-user compatibility check. No problems were found with the ECHO program.

The SINK program is a server which simply discards data sent to it. It is intended to be used in measuring bandwidth and delay over TCP connections. To date, SINK has been used only enough to verify that it is functional.

The most-used TENEX server is TTLSRV, a Telnet-like server which permits a remote system to make a teletype connection to TENEX and actually "talk" to a job on TENEX. Testing of this server uncovered several bugs which were causing the data stream to be garbled. Once repaired, TTLSRV was used from a remote location in California via the San Francisco Bay Area Packet Radio Network. The terminal was an LSI-11 computer and the ARPANET was accessed through a gateway in the PRN Station computer.

The TCP and all three servers are now in everyday operation. They are automatically started when TENEX is brought up. Log files are kept to monitor their use. These programs are running in their own 5% pie slice on BBNA so that an estimate of the CPU requirements may be gained during the ongoing tests.

C. Gateway Development

At the beginning of this quarter, the gateway software for the ARPANET/Packet Radio Net had been coded and debugging in the BBN Packet Radio Net was in progress. In the early part of this quarter, the gateway was modified to use the new style internet header formats which are now the standard in internetworking

experiments. This gateway software was delivered to SRI and functionally demonstrated in the San Francisco Packet Radio Network at the end of July.

Subsequent to this delivery, we began working on a new version of the gateway software to be used in the ARPANET/Satellite Net gateway. This version includes revisions of the gateway software, the addition of "fake hosts," and the addition of network specific code in the gateway for supporting communications on the Satellite Net.

At this time, the gateway software, that is the programs which re-address packets between networks, was redesigned in a more modular form, in order to facilitate coding of additional gateways planned for the Satellite Net. To provide measurement and testing facilities in the gateways, we designed the gateway to support a number of "fake hosts." These are application programs which can be run on the gateway machine, interfaced to the gateway software. A general interface to the gateway which can be used by any fake host to communicate on any network accessible from the gateway was designed and implemented. This interface provides a mechanism for identifying the fake host to the gateway and opening or closing inter-process communications between the gateway and fake host for transferring data to and from the fake hosts and the networks.

In addition to the fake hosts and the programs which re-address packets between networks, a gateway machine includes a set of network specific programs which are used to support communications on the particular networks to which the gateway is attached. Communications between the gateway and the hosts and IMPs on the ARPANET is supported by the XNCP, an experimental network control program which is incorporated in the ELF operating system run in the gateway machines. As the Satellite Net protocols are currently the same as the ARPANET protocols, this program was expanded to support communications on both the Satellite Net and the ARPANET. The ELF software also had to be modified to support the very distant host interface to the Satellite Net. Although a VDH driver and a Reliable Transmission Protocol (RTP) package did exist for the ELF system, these had to be modified to interface with our version of the XNCP. This required considerable rewriting of the RTP package, as this was originally coded to interface to the standard ELF Network Control Program.

At the end of the quarter, debugging of the revisions to the RTP package and the interface to the XNCP is in progress. The coding and debugging of the gateway software for the ARPANET/Satellite Net gateway was near completion and the gateway was being used to debug a set of fake hosts which will provide traffic generation and measurement facilities in the gateway.

V. HARDWARE

During this quarter the second Packet Radio Station PDP-11 computer was delivered and checked out. The machine was installed and found to operate properly.

The ANTS interface was installed in the new PDP-11, but did not operate reliably in the present configuration. Two possible causes of difficulty were identified: (1) the interface is now connected to a Pluribus machine, instead of to an ARPANET IMP, and (2) the cable to the interface is now a 400 foot cable. The different timing and cable characteristics seemed sufficient explanation of the errors noted, and improved circuitry in the interface was installed to alleviate the problem. The present remedy is only marginally adequate, however, and we plan to replace the interface with a standard IMP11-A interface. A proposal to procure an IMP11-A for this purpose was submitted during this quarter.

This quarter also saw the arrival of a Very Distant Host (VDH) interface. This will connect the gateway PDP-11 computer to the ARPANET. Actually, delivery of the VDH involved arrival or three distinct interfaces; the first was damaged in shipment, the second made intermittent errors, and the third is now installed and in use. Overhead in dealing with the first two interfaces has delayed our use of the VDH, although the supplier has been

very cooperative in helping us diagnose problems and ultimately obtain a working unit. Cables and modems have been installed which connect the VDH to a spur of the ARPANET for testing and initial software development. The VDH has been tested in local busback mode and found to work. Work during the next quarter will involve testing the VDH connection to the ARPANET and developing interface driver software for inclusion in gateway routines.

TABLE OF CONTENTS

I. INTRODUCTION

During the past quarter, we developed variable frame rate schemes for transmitting pitch and gain for use with ARPA LPC-II speech compression system. Employing these schemes, the average transmission rate of LPC-II is expected to be about 2000 bps for continuous speech. We have just issued ARPA NSC Note 96 providing specifications of these transmission schemes so other ARPA-sponsored sites can incorporate them into their LPC-II implementations.

We successfully completed the first phase of our perceptual modeling work: In this phase, 1) we developed an interactive display program on our PDP10/IMLAC PDS-1 (display minicomputer) facility; 2) we incorporated into this program a number of features which allow us to manually mark selected frames of analyzed data for transmission, synthesize speech from a specified amount of transmitted data, and play out through a D/A converter specified portion of either synthesized speech or natural speech or both for "on-line" evaluation of transmitted data in terms of relative speech quality; and 3) for several utterances, out of the available 100 frames/sec LPC data produced by the analyzer, we manually selected an average of about 30 frames/sec data using some simple rules in such a way that the resynthesized speech was almost indistinguishable from the speech synthesized with all the 100 frames/sec analysis data transmitted.

We modified the excitation part of our LPC synthesizer program to permit the use of one of the following three excitation models: 1) periodic pulse/random noise excitation, which is the one we have exclusively used in the past; 2) all-pass excitation, which is obtained by passing the pulse/noise excitation signal through an all-pass network; and 3) linear prediction residual or error signal. Reasons for considering options 2) and 3) are given later in Section IV.

Our work on real-time LPC implementation was directed towards developing a real-time interface (A/D and D/A facility) for our PDP-11/SPS-41 system.

In speech quality evaluation we generated the stimulus tapes, and collected rating data in a new subjective quality evaluation study, and we are currently analyzing the results. The purpose of the study was both to provide baseline data for our objective measures of quality, and to establish the optimal combination of vocoder parameter values (number of coefficients or poles used in LPC analysis; step size for quantizing these coefficients; and number of frames of LPC parameters transmitted per second), that would yield the best quality for a given transmission rate in bits per second.

II. VARIABLE FRAME RATE TRANSMISSION OF PITCH AND GAIN

In the beginning of this year, we provided specifications for ARPA LPC-II speech compression system (see our NSC Note 82) in which a variable frame rate (VFR) scheme is used for transmitting log area ratios (LARs) and essentially a fixed rate transmission is used for pitch and gain; we estimated the average transmission rate of LPC-II as 2200 bps for continuous speech. Last quarter, we developed VFR schemes for transmitting pitch and gain also at a variable rate. We have documented the details of these schemes along with our recommendations for their implementation in LPC-II in the recently-issued NSC Note 96; a summary of the results was also presented at the August ARPA NSC Meeting. A copy of this note is reproduced here as the Appendix, so we will not give any details in this section. Employing these VFR schemes, the average transmission rate of LPC-II is expected to be about 2000 bps, which represents a saving of 200 bps over the bit rate of 2200 bps obtained with a fixed rate transmission of pitch and gain. Through informal listening tests on the quality of speech from our LPC simulation system, we found that this saving in bit rate was achieved without causing any noticeable change in speech quality.

III. PERCEPTUAL MODELING

The basic assumption underlying our perceptual model is that continuous speech can be represented in terms of LPC parameters extracted at a minimal set of perceptually significant time points, not necessarily equally spaced. In the context of a speech compression system, these time points correspond to instances or frames when transmission of LPC parameters must occur. Thus, the goal of our perceptual modeling task is to develop a perceptually based variable frame rate transmission scheme for LPC parameters, and use it instead of the presently employed likelihood ratio method. We expect that this approach would noticeably enhance the vocoded speech quality.

We set out to perform the perceptual modeling task in two phases. In the first phase, which was completed in the last quarter, our goal was to manually select a minimum number of frames of parameter data for transmission based on the information about these parameters only, and in such a way that the resynthesized speech was almost indistinguishable from the speech synthesized with all the analysis frames of data transmitted. The second phase of work is to automate the manual selection procedure, and we expect to accomplish this in the current quarter.

A. Interactive Display Program

As a key tool for carrying out the perceptual modeling task, we developed an interactive display program on our PDP10/IMLAC PDS-1 facility. The program displays all the transmission parameters (LARs, pitch and gain), as well as the transmission status (0 or 1) of each of these parameters for every analysis frame, as functions of time (or frame number). By setting appropriate hardware switches, the program can be made to display the values of displayed parameters for the frame indicated by the current position of the display cursor, and/or the spectrum of the linear prediction filter for that frame, and/or the speech waveform in that frame. (The underlying display format of this program was originally developed as part of our Speech Understanding Project.)

By viewing the displayed information for several utterances, one gains an intuitive feel for the magnitudes of parameter variation under various speech events and starts to develop simple rules that may be used in deciding whether or not a given frame of data should be transmitted. To further aid the user, we incorporated a number of features which allow the user to

1. manually mark selected frames of analyzed data for transmission,

2. synthesize speech from a specified amount of transmitted data, and

3. play out through a D/A converter specified portion of either synthesized speech or natural speech or both for "on-line" evaluation of transmitted data in terms of relative speech quality.

The user can choose to save in disk files (for use in future sessions) the positions of transmitted frames (or transmission marks, for short), and the samples of synthesized speech.

We used these features extensively in developing and evaluating transmission criteria, and in creating manually-derived transmission mark files along with the corresponding synthesized speech files for several utterances, so they can be used later for comparing with the results obtained using automatic transmission schemes.

B. Manual or Non-automatic Perceptual Modeling

Using the above interactive program, we accomplished the task of manually deriving the perceptual model for several utterances. LPC analysis was done to extract pitch, gain and 14 log area ratios at a rate of 100 frames/sec, or once every 10 ms, from speech sampled at 10 kHz. We selected a minimum number of frames of data for transmission, out of the available 100 frames/sec analysis data, using only the information about the transmission parameters and employing rules such as the following:

1. when log area ratios change roughly linearly, transmit them only for the frames corresponding to the endpoints of the line, since the LARs for the intermediate frames will be generated at the receiver through linear interpolation, and

2. ignore or deemphasize large changes in the values of LARs when the associated filter gain is low, since these low-gain frames have a relatively small effect on perception;

the overall objective was to reduce the frame rate as much as possible with the constraint that the resynthesized speech should be almost indistinguishable (as judged by informal listening tests) from the speech synthesized with all the analysis frames of data transmitted. We achieved a minimum frame rate of about 30 frames/sec on the average, which represents more than a 3-to-1 reduction from the available 100 frames/sec analysis data. In this investigation, speech signal was not preemphasized; transmitted parameters were not quantized; and, pitch and gain were transmitted at the analysis rate of 100 frames/sec. Work is in progress to study the effect of parameter quantization. Based on our recently completed work on variable frame rate transmission of pitch and gain (see Section II and Appendix), we can achieve an average transmission rate of about 35 frames/sec without noticeable change in speech quality.

IV. SYNTHESIZER EXCITATION

In the past, we have exclusively used, as excitation or input signal for the synthesizer filter, periodic pulse sequence for voiced frames and random noise sequence for unvoiced frames. The difficulty with using this pulse/noise excitation for the minimum-phase LPC synthesizer is that the synthesized speech has larger peak amplitudes than the natural speech used in the analysis. To accommodate this situation, we used 9 bits to store input or natural speech samples, and 12 bits to store synthesized speech samples. Since the full dynamic range possible with 12 bits was not effectively used in storing the synthesized speech samples, the signal-to-noise (noise at the D/A converter) ratio was lower, producing sometimes less desirable audio quality at the output of the D/A converter. To overcome this problem, we modified the synthesizer program to allow the use of an all-pass excitation.

A. All-Pass Excitation

We chose an 8th order all-pass filter given in [5], which was specifically designed to minimize the peak amplitude of its impulse response. All-pass excitation signal can be obtained by filtering the pulse/noise excitation signal through this all-pass filter. To simplify computations, however, we precompute once at the start 40 samples (4 ms at 10 kHz sampling rate) of the impulse response of the all-pass filter and store them in memory. If a given frame is

unvoiced, we use the random noise sequence directly as the excitation signal (i.e., no all-pass filtering is done); this strategy is fine since high peak amplitudes occur only in voiced speech. For a voiced frame, we choose one of the following two cases, depending on the value of the pitch period for that frame: 1) If pitch period is longer than 4 ms, we take the 40 samples of the all-pass impulse response and append at the end with the required number of zeros to generate the excitation signal. 2) If pitch period is shorter than 4 ms, we use the "aliased" version of the 40-sample sequence which is obtained by considering the periodic occurrence of this sequence at a rate given by pitch frequency.

By conducting synthesis experiments, we found that peak amplitudes were in fact lowered when using the specific all-pass excitation discussed above. Even in this case, however, peak amplitudes of synthesized speech were higher than those of the natural speech; the increase in peak amplitudes due to synthesis was often found to be about 6 dB or less. We accommodate this increase by using 11-bit natural speech samples, and 12-bit synthesized speech samples. Using this approach, the audio quality of speech at the output of the D/A converter was found to be better than what we had previously found. We already used this approach in generating stimuli for subjective quality tests described in Section VI.

B. Residual Excitation

For this case, the linear prediction residual or error signal is used as the excitation signal for the synthesizer. Of course, one needs a large bandwidth for transmitting the error signal; the overall transmission rates of LPC vocoders that transmit the error signal range from about 8 kbps to about 16 kbps depending on the employed sampling rate and number of bits/error sample used for its quantization. The reason for incorporating residual excitation in our program is to be able to compare the speech quality of a residual-excited vocoder with that of a pitch-excited vocoder under different choices of vocoder parameters such as quantization step size, number of poles and frame rate. A pilot study involving a small number of such comparisons was included in the subjective experiment described in Section VI.

V. REAL-TIME IMPLEMENTATION

Our effort in the last quarter was directed towards developing a real-time interface (A/D and D/A facility) for our PDP-11/SPS-41 system to replace the one on our PDP-10 (System D) since the latter will be returned to DEC in the near future. To this end, we now have a usable RT11 operating system on the PDP-11, and a preliminary version of FTP (File Transfer Protocol) which runs under RT11.

VI. SPEECH QUALITY EVALUATION

In the last QPR (No. 6), we outlined a preliminary design for a subjective rating study of 82 vocoders. Its main purpose was to collect reliable subjective data for use in developing and calibrating our objective measures of quality. Other aims were to determine 1) the combinations of vocoder parameters yielding best speech quality, for a variety of bit rates, 2) the additional savings in bit rate achievable by variable transmission rate, without further quality degradation, and 3) the extent to which inadequate modeling and coding of the excitation source contributed to reducing the quality and naturalness of the vocoded speech. We have since decided that the compromises in experimental designs that were required, if all of the foregoing aims were to be met in a single experiment, were not outweighed by the benefits. Therefore, we decided to split up the experiment into three smaller ones, each directed to one of the three aims.

The data collection for the first of these is now complete, and we are in the process of analyzing the results. The details of the final design and the procedures we used for collecting the data are described below.

A. The Purposes of the Study

As described in our earlier QPR's, we are trying to develop objective measures of vocoder quality, which can predict and therefore replace the present subjective testing methods, which are both costly and time consuming. Reliable subjective data is needed both to compare alternative objective measures, and also for suggesting possible ways of improving the predictive power of the measures. Providing this reliable subjective data is a major purpose of the new study. The main constraints that this purpose imposes on the study are 1) that a wide range of speech qualities should be represented in the systems studied, and 2) that there should also be systems of equivalent quality, but in which the degradation is due to differences in the choice of the vocoder parameters.

The second purpose was to establish which combinations of vocoder parameters result in the highest quality speech, for a variety of overall bit rates. In general, the quality of LPC vocoded speech declines monotonically as the bit rate is reduced, regardless of which parameter is manipulated to reduce the bit rate. But the rate at which quality declines depends both on which parameter is manipulated, and the particular value of the parameter.

To establish the best operating point, for a range of different bit rates, it is therefore necessary to perform a factorial study.

In a factorial study, each value of a parameter of interest occurs with every combination of values of the other parameters. From informal listening tests, we suspected that eleven poles were needed, in order that male voices not sound muffled (see QPR No. 4); that the step size for log area ratio quantization could be as large as 1 dB; and that about 50 frames of data should be transmitted per second. Therefore, we used the following set of parameter values, in all combinations, yielding 48 LPC systems (4x3x4).

Parameters	N	Values
No. of Poles	4:	13, 11, 9, 8
Quantization Step Size	3:	0.5, 1.0, 2.0 dB
(Fixed)Frame Rate	4:	100, 66-2/3, 50, 33-1/3 frames/sec

Two additional systems were included: one an LPC system with 13 poles, 0.25 dB quantization step size, and transmission rate of 100 frames per second, which gives it an overall bit rate about 20% higher than any of the 48 systems, and the other corresponding to 110 kbps PCM (i.e. the waveform sampled at 10 kHz and quantized to 11 bits), to act as an undegraded anchor. The overall bit rate of the LPC systems, including pitch and gain, ranged from 8700 bps (13 poles, 0.25 dB quantization, 100 frames/sec) down to 1267 bps (8 poles, 2.0 dB quantization, 33-1/3 frames/sec). The bits per frame for each combination of number of poles and quantization step size

are given in Table 1. The overall bit rate for any system is calculated by adding 11 bits of pitch and gain coding (6 bits for pitch and 5 bits for gain) to the bits per frame, and multiplying by the appropriate frame rate. Note that the bit rate so obtained does not include the benefits of Huffman coding, in which the most frequently used values are assigned the shortest codes. This procedure can further reduce bit rates by about 20%, with absolutely no change in the coefficient values transmitted.

Table 1

Quantization Step Size	No. of Poles 13	11	9	8
0.25 dB	76	--	--	--
0.5 dB	63	55	47	43
1.0 dB	50	44	38	35
2.0 dB	37	33	29	27

Table 1: Bits per frame for all combinations of number of poles and quantization step size used in the present study. Overall bit rate is obtained by adding 11 bits of pitch and gain coding to the given values, and multiplying by the appropriate frame rate (100, 66-2/3, 50, or 33-1/3 frames/sec).

B. Choice of Sentence Materials

The results of our earlier subjective quality tests, whose purpose was to develop a testing method, showed clearly that, if meaningful results are to be obtained from preference ratings, all sentence materials must be passed through all systems (See QPR No. 4). Other researchers have reached similar conclusions [4]. If a different randomly determined subset of the material is passed through each system, an implicit assumption is being made that the speech material is homogeneous. Our results showed that the relative quality of speech processed by two vocoders can be reversed, for two different speech samples, or for two speakers with different voice characteristics. Therefore, the assumption that speech is homogeneous is not warranted.

In our earlier tests, we developed a set of six sentences, each read by six talkers, that was both representative, in that it covered a wide range of speech events and talker characteristics, and also challenging, in that some speech material was included that would fully extend any LPC vocoder's abilities. The rationale governing generation of the material was given in detail in QPR No. 1. Unfortunately, we could not use all 36 speaker-sentence combinations in the present study, since passing them through all 50 vocoder systems would have made the study unmanageably large. We therefore used the plotted results of our earlier experiments (QPR No. 4, Appendix) to select a subset of seven speaker-sentence

combinations. The subset was chosen to include: each of the six sentences at least once, and five of the six speakers at least once. (The sixth speaker was excluded because she had a regional accent, and also spoke rather slowly). About half of the subset was to be spoken by males and half by females. The vectors representing the seven selected sentences, in the perceptual space built by multidimensional scaling analysis of the data from our earlier experiment (QPR No. 4), had to be as widely separated as possible to ensure that the solution generated for the subset would be highly similar to that generated for the full set of 36 speaker-sentence combinations.

The subset of sentences that best met all of the foregoing constraints consisted of: JB1, DD2, RS3, AR4, JB5, DK6, and RS6. Relevant details of the sentences, and of the speakers voices, are given in Table 2.

To confirm that the subset was, in fact, representative of the whole set of 36 speaker-sentence tokens, a multidimensional analysis was performed, using MDPREF (See QPR No. 2), on the subset of rating data obtained for the seven selected sentences processed by the 14 vocoder systems used in our earlier subjective rating study. The solution obtained for the subset was highly similar to the solution presented in QPR No. 4 for the complete set of speaker-sentence combinations.

C. Generation of Stimulus Tapes

The total set of stimulus materials consisted of the seven sentence tokens, each passed through all 50 vocoder systems, to yield a total of 350 different stimulus items. The first step in generating the stimuli was to generate the computer waveform files, each corresponding to an input sentence waveform, transformed by processing through a simulated vocoder with the desired parameter settings. The 350 waveform files were then played out and recorded on a master tape. From there, the stimuli were dubbed onto Language Master cards, to permit random access.

Earlier studies have demonstrated that a subject's judgment, especially of speech stimuli, can be strongly affected by the preceding stimulus (e.g. [2]). For example, a judgment on a mid-range stimulus tends to be moved towards "bad" when it is preceded by a clearly superior stimulus, and towards "good" when preceded by an inferior stimulus. It is important to control for this effect by counterbalancing the presentation order. A complete counterbalancing, in which every stimulus followed each other stimulus exactly once, would have required 350 passes through the 350 stimuli, which is neither practical nor necessary. We therefore decided to produce a complete counterbalancing of the 50 vocoder systems, and an independent approximate counterbalancing of the sentences. Such a counterbalancing required only seven passes through the 350 stimuli, and had the further advantage that even

TABLE 2

#	ID	M/F	av fØ 50%	Sentence	Descriptor
1.	JB1	M	119 ± 20%	Why were you away a year, Roy?	Vowels and Glides
2.	DD2	M	134 ± 15%	Nanny may know my meaning.	Nasals
3.	RS3	F	195 ± 42%	His vicious father has seizures.	Fricatives
4.	AR4	F	165 ± 32%	Which tea-party did Baker go to?	Transients
5.	JB5	M	124 ± 20%	The little blankets lay around on the floor.	General
6.	DK6	M	97 ± 13%	The trouble with swimming is	
7.	RS6	F	193 ± 26%	that you can drown	General

within each pass, all ranges of contrast between successive stimuli occurred equally often, so that no severe departures from balance occurred even within one pass. The sequence was generated by a trial and error search, following an algorithm described in [6]. No system and no sentence was preceded by itself.

Seven experimental tapes were then recorded. The 350 Language Master cards were sorted into the correct order for one pass, and dubbed onto the stimulus tape in blocks of ten. Stimuli within one block occurred at a rate of one every 7.5 seconds, with a longer gap between blocks. (This stage of generation of stimulus tapes will be greatly simplified and shortened when the tape can be made on-line, as will soon be possible with our RT-11 real-time system).

In addition to the protection from sequence effects provided by counterbalancing the presentation order, we tried to further reduce sequence effects (and thus improve the reliability of the data) by a novel method. Since auditory stimuli are presented sequentially rather than simultaneously, sequence effects must be the result of the present stimulus being compared with the memory-trace of the preceding stimulus. Anything that weakens the memory-trace must reduce the sequence effect. In studies of short term memory, it has been shown that memory for the last item in a list is substantially reduced if it is followed by a redundant "suffix", a final list item that the subject knows, and does not have to report, see [1]. The redundant suffix seems to interfere with the "precategorical" (i.e.

sensory) memory trace of the preceding stimulus. We used this effect to minimize sequence effects. About one second after the end of each stimulus sentence, a continuous speech babble was faded in, at about the same perceived level as the stimulus. The babble was faded out again about a second before the start of the next stimulus. The babble was faded automatically by a photo-electric isolator, whose light supply was interrupted whenever a card bearing a stimulus sentence was played through the Language Master. The babble consisted of six voices, each reading a continuous text passage, mixed under computer control to ensure a highly regular signal. The babble was developed at BBN in a separate project, described in more detail in [3]. The babble signal was recorded on the second track of the tape, with the stimuli on the first track, to permit the signal to be played with or without the babble.

D. Experimental Procedures

The subject's task was to rate the degradation of the stimuli he heard. This negative attribute was chosen for sealing, as in our earlier experiment, because the scale has a natural origin, or zero, corresponding to undegraded speech. Scales which assign larger values to positive attributes of the speech, such as scales of quality, clarity, acceptability, etc., do not have this advantage, and ceiling (or floor) effects may occur at both ends of these scales.

Instead of assigning a number to his judgment, the subject made his response by making a mark on a 10 cm line on his answer sheet. We adopted this method of obtaining the responses because we felt that subjects have highly idiosyncratic ways of using numbers, even when carefully instructed -- as evidenced by the different response histograms obtained in our last rating study (See QPR No. 3, Part III, Figure 1). Forcing a subject to use a system for assigning numbers to his judgments that he would not have chosen, can considerably degrade his performance. Furthermore, it is always difficult to decide how many categories of response to allow. If the number is too small, the subject has to classify stimuli as the same, even though he hears them different. This both throws away useful information and upsets the subject. The ideal number of categories may vary from one subject to another. If too many categories are used, this confuses the subject and reduces his confidence in his judgments. Obtaining responses on a continuous scale, as we did, means that any suitable categorization can be imposed on the data later. Two visual anchors were provided on the response line. The left anchor was 4 mm from the left end of the line, and was marked "PERFECT". The right anchor was 1 cm from the right end of the line. Each page of the response booklet carried ten of these "response scales", as shown in Figure 1. The subject covered his earlier responses with a card as he moved down each page, to reduce response bias. For computer entry of the data, the continuous response variable was converted into the distance in

millimeters from the left end of the line (not the anchor) to the subject's mark where it crossed the response line.

Nine subjects have served so far in the experiment. We plan to analyze their data, and run more subjects if the results are not sufficiently reliable for our needs. The subjects were recruited by local university summer placement offices: all subjects reported having normal hearing.

Three of the subjects made the first five passes through the 350 stimuli, and six more subjects made only the first two passes. The experiment took three afternoons, the last two for data collection, and the first for instructing and training the subjects.

The schedule of stimulus presented is shown in Table 3. A complete pass through the 350 stimuli was presented in 35 blocks of 10 stimuli. On the first day of data collection, 120 blocks of stimuli were presented. Rests were allowed after every 20 blocks. After the first 60 blocks were presented, blocks 1-20 were repeated. Comparison of responses on the second pass through these blocks with those on the first pass indicates whether subjects were consistent in their responses, in that the relative ratings of the various systems was the same in the two passes, and also whether their responses have drifted or not, as reflected by the ratings for the two passes having similar absolute values.

Repeated judgments were also obtained on blocks 61-70, one repeat with a separation of only 10 blocks, a second over the weekend intervening between the two data collection days, and a third over the second data collection day. Subjects were not aware that any blocks were repeated. (The word "Pilot" in Table 3 refers to a pilot experiment we included in the main experiment to give an idea of how much degradation is due to inadequate modeling and coding of pitch and gain. The results will be described in the next QPR.)

The data have all been entered into computer files, and checked, and analysis of the data is proceeding. We will postpone presentation of any of the results until the analysis is complete.

TABLE 3

Schedule of Stimulus Presentation

Sentence Blocks	Tape #	Response Booklet #
Day 1: 9 Subjects		
1-20	1	1
21-40	2	2
41-60	2	3
1-20	1	4
(refreshment break)	---	---
61-80	3,4	5
61-70 and Pilot	3	6
Day 2: 3 Subjects		
61-70	3	7
81-100	5	8
101-120	5	9
121-140	6	10
141-160	6	11
(refreshment break)	---	---
161-175	7	12
61-70 and Pilot	3	13

REFERENCES

1. Crowder, R.A and Morton, J. (1969). Precategorical Acoustic Storage. Perception and Psychophysics 5, p. 365-375.

2. Huggins, A.W.F. (1968). The Perception of Timing is Natural Speech I: Compensation within the Syllable. Language and Speech, 11, p. 1-11.

3. Kalikow, D.N., Stevens, K.N. and Elliot, L.L. (1976). Development of a Test of Speech Intelligibility in Noise Using Sentence Materials with Controlled Word Predictability. BBN Report No. 3370.

4. Pachl, W.P., Urbanek, G.E. and Rothauser, E.H. (1971). Preference evaluation of a large set of vocoded speech signals, IEEE Trans. Audio-Electroacoustics, AU-19, p. 216-224.

5. Rabiner, L.R. and Crochiere, R.E. (1976). On the Design of All-Pass Signals with Peak Amplitude Constraints. Bell System Technical Journal, Vol. 55, p. 395-407.

6. Williams, E.J. (1950). Experimental designs balanced for pairs of residual effects. Australian Journal of Scientific Research A3, p. 351.

APPENDIX

VARIABLE FRAME RATE TRANSMISSION OF PITCH AND GAIN

NSC Note 96, September 30, 1976

(Author: R. Vishu Viswanathan)

I. INTRODUCTION

This note specifies schemes for variable frame rate (VFR) transmission of pitch and gain for use with ARPA LPC-II speech compression system [1]. Employing these VFR schemes, the average transmission rate of LPC-II is expected to be about 2000 bps for continuous speech, which represents a saving of about 200 bps over the bit rate of 2200 bps for the system specified in [1] in which essentially a fixed rate transmission of pitch and gain was considered. Through informal listening tests on the quality of speech from our LPC simulation system, we found that this saving in bit rate was achieved without causing any noticeable change in speech quality.

II. DESCRIPTION OF VARIABLE FRAME RATE SCHEMES

Below, we consider the same type of VFR scheme for both pitch and gain. For convenience, we use the word parameter to mean either pitch or gain. A simple VFR scheme transmits the parameter whenever it changes by more than a prespecified amount (threshold) since the last transmission. A step-by-step description of this single-threshold VFR scheme is given below, where T denotes the preselected threshold.

(1) Transmit value at frame n
    $i \leftarrow 0$

(2) $i \leftarrow i + 1$
    $D \leftarrow |(\text{frame } n+i \text{ value}) - (\text{frame } n \text{ value})| - T$

(3) If $D \leq 0$, go to (2). (No transmission).

(4) $n \leftarrow n + i$, go to (1).

To avoid having to do parameter interpolation between largely different parameter values at the receiver, as in [1] we recommend using a double-threshold VFR scheme described in the next page, where T1 and T2 are the two preselected thresholds, with T2 > T1. In words, if the magnitude of the change in parameter value between a current frame and the previously transmitted frame exceeds only T1, and not T2, then the current frame is transmitted; if it exceeds both thresholds, then the parameter for the frame

immediately preceding the current frame is transmitted.

(1) Transmit value at frame n
    i ← 0

(2) i ← i + 1
    D1 ← |(frame n + i value) - (frame n value)| - T1

(3) If D1 ≤ 0, go to (2). (No transmission).

(4) If i = 1, go to (7)

(5) D2 = |(frame n + i value) - (frame n value)| - T2

    If D2 ≤ 0, go to (7)

(6) i = i - 1

(7) n ← n + i, go to (1)

In the description of either of the above two schemes, by the phrase 'frame n value' we mean either (1) some predefined function of the parameter value before quantization at frame n or (2) the (integer) quantization level corresponding to the parameter value at frame n. For case (1), we exclusively considered the logarithm of the unquantized parameter value as follows:

Pitch, P: $\log_{10} P$,

Gain, G: $10 \log_{10} G$ (decibels),

where P is pitch period in number of samples (or equivalently pitch frequency in Hz, since the above VFR schemes consider only the magnitude of the differences in frame values), and G is average energy per sample of speech over the analysis interval. With case (2) above, the VFR scheme can be located after the quantizer, so it can be viewed as part of the encoder. Clearly, the VFR scheme that employs quantized levels is simpler to implement since it does not require the computation of logarithms of parameter values. In addition, as we shall see later, for a given frame rate of transmission, the two cases result in about the same speech quality. Based on these reasons, we recommend the VFR scheme that employs quantized levels. In the next section, however, we present experimental results for both cases (1) and (2), for comparison purposes.

III. EXPERIMENTAL RESULTS

We used our floating-point simulation of the LPC vocoder for obtaining the results stated below. Briefly, we employed 10 kHz sampling rate, 9-bit A/D converter, and pitch extraction using the center-clipping method. Also, pitch frequency had a range of 50 Hz-450 Hz, while the gain parameter G had a range of 45 dB. We computed pitch and gain once every 10 ms, and our data base consisted of 11 sentences of total duration of about 25 sec. In spite of the differences between our simulation system and LPC-II, we expect the results given below for the VFR transmission of pitch and gain to hold approximately for the conditions of LPC-II.

Considering pitch, plots of average frame rate of transmission versus threshold are shown in Fig. 1 for two situations, both using unquantized log pitch for deciding transmission: The lower plot is for the single-threshold VFR scheme, while the upper plot corresponds to a double-threshold VFR scheme with the sum of the two thresholds being a constant along that plot; for the latter plot, abscissa gives the value of the first threshold. Similar plots are given in Fig. 2 for the case where the VFR scheme employs quantized pitch levels. (As in [1], we used 6 bits for pitch quantization.) Of course, for the case shown in Fig. 2, the number and duration of individual unvoiced regions directly affect the frame rate of pitch

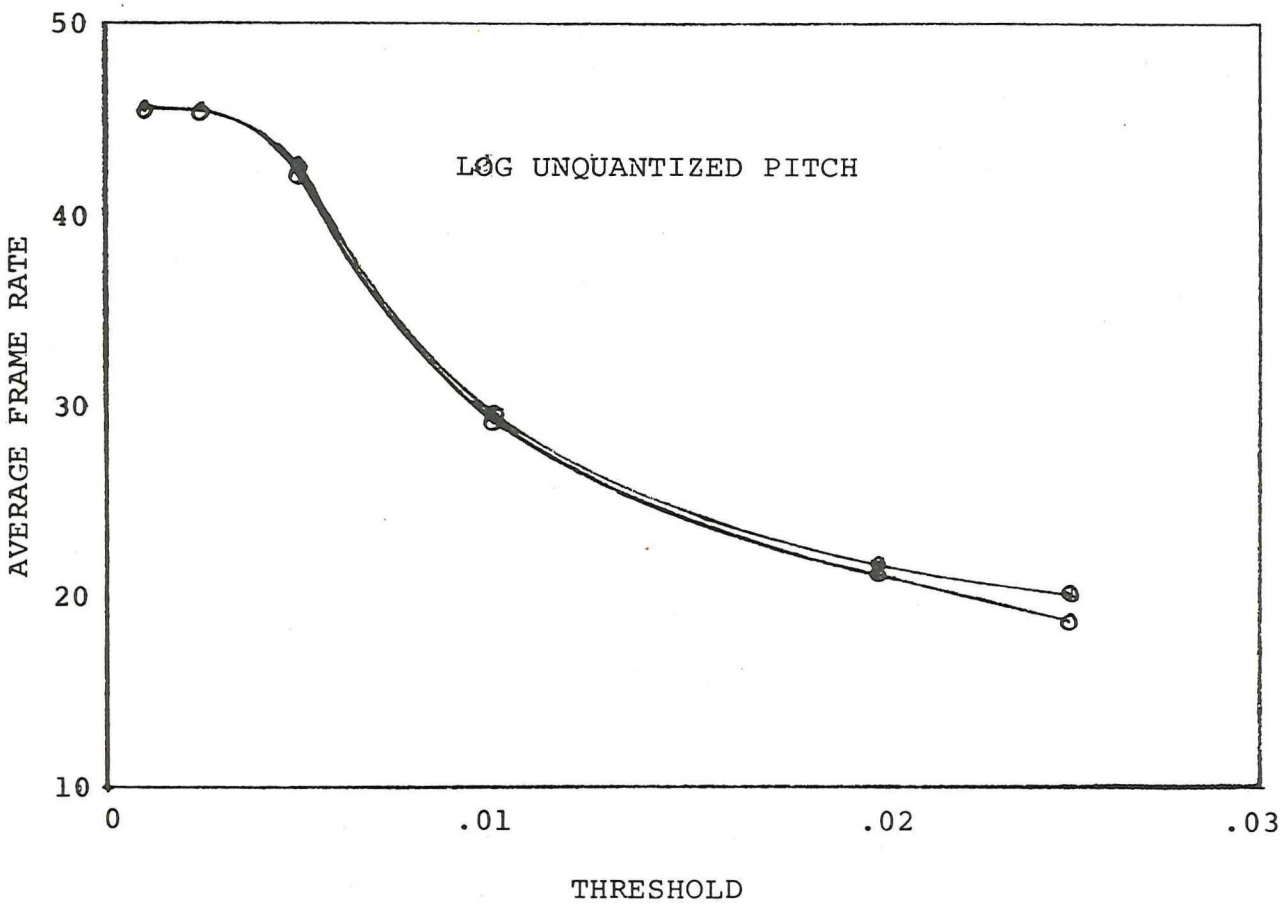

Fig. 1. Average frame rate of pitch transmission as a function of threshold value employed in the VFR scheme which considers change in logarithm of unquantized pitch P for deciding when to transmit.

Lower curve: Single-threshold VFR scheme
Upper curve: Double-threshold VFR scheme with the sum of the two thresholds P1 and P2 kept constant at 0.06; P1 is plotted along the X-axis in this case.

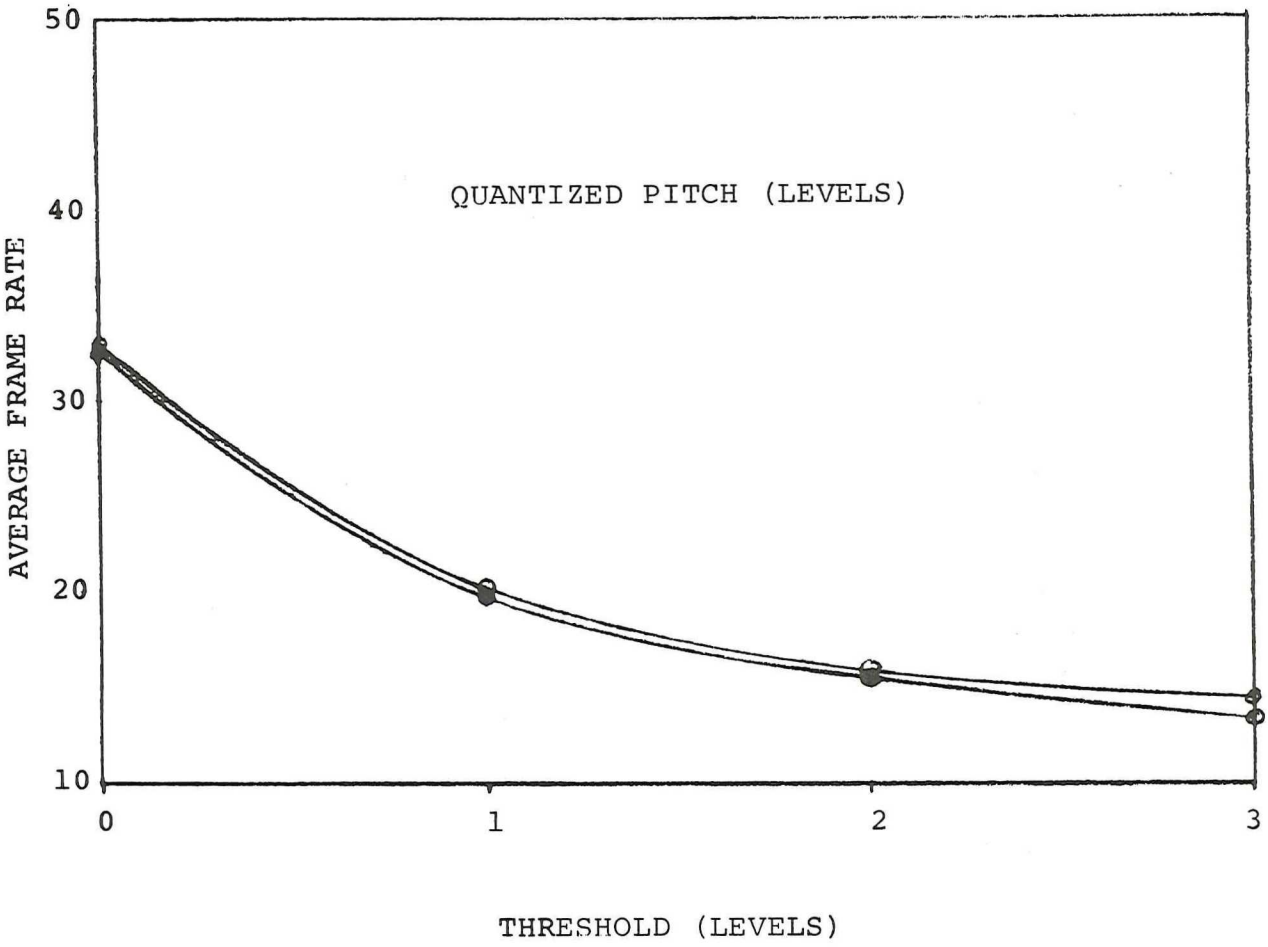

Fig. 2. Average frame rate of pitch transmission as a function of threshold value employed in the VFR scheme which considers change in quantized pitch level IP for deciding when to transmit.

Lower curve: Single-threshold VFR scheme
Upper curve: Double-threshold VFR scheme with the sum of the two thresholds IP1 and IP2 kept constant at 7; IP1 is plotted along the X-axis in this case.

transmission since only the first pitch value in each unvoiced region is transmitted. The data base we used for computing average frame rate was continuous speech except for short silences (less than 200 ms duration) at the ends of each sentence; the total unvoiced duration (including silences) was about 30% of the total duration of the data base.

From Fig. 2, we see that a threshold of zero already reduced the transmission rate from 100 fps (frames/sec) to about 33 fps. Notice that a zero threshold means that pitch is transmitted whenever its quantized level is not equal to that of the last transmitted pitch; thus, the receiver has the same pitch data as that at the quantizer output. That we can reduce the pitch frame rate from 100 fps to 33 fps without introducing any approximation is a significant and useful result.

For the VFR scheme using quantized pitch, Fig. 3-a shows the transmission interval histogram which is the plot of percentage of total transmitted frames versus interval between adjacent transmissions, while Fig. 3-b displays percentage savings in frame rate as a function of transmission interval.

Considering the VFR transmission of the gain parameter, the unquantized and quantized cases are shown plotted in Fig. 4 and Fig. 5, respectively. For the quantized case, the transmission interval histogram and percentage frame

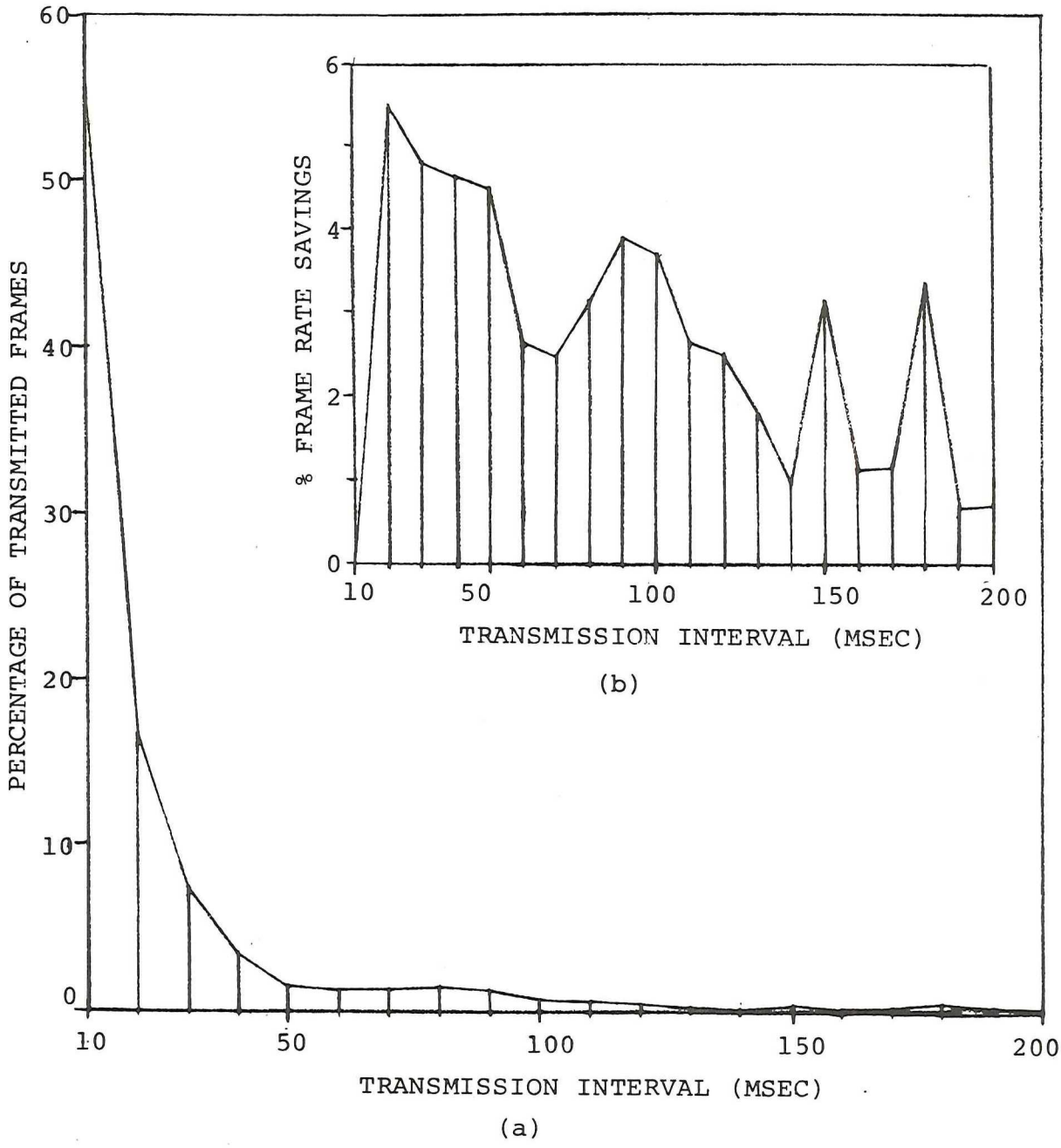

Fig. 3. Double-threshold VFR scheme for quantized pitch levels with the two thresholds held at IP1=0, IP2=25.

Plot (a): Histogram of transmission interval

Plot (b): % Frame rate savings versus transmission interval

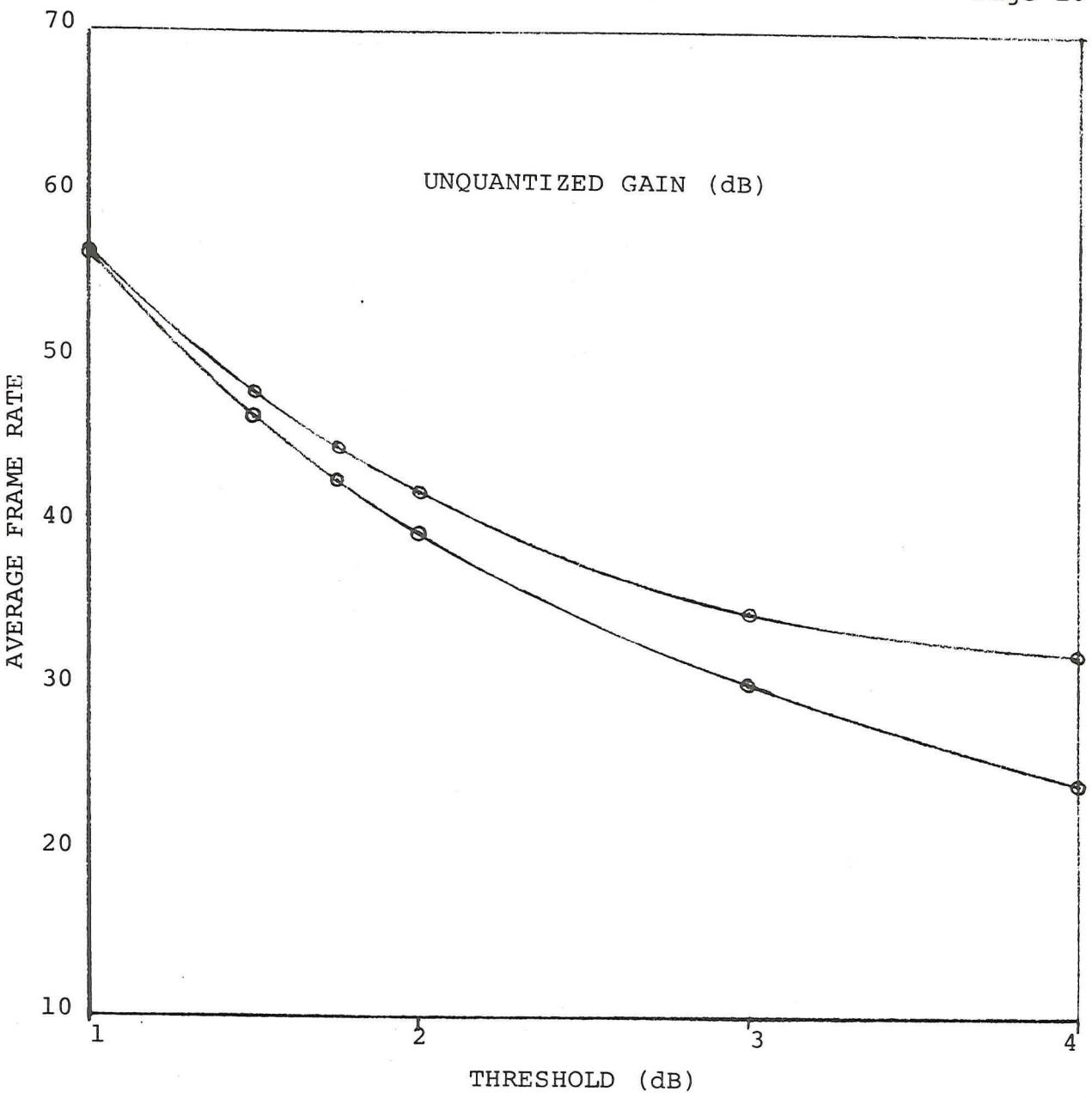

Fig. 4. Average frame rate of gain transmission as a function of threshold value employed in the VFR scheme which considers change in unquantized gain G in decibels for deciding when to transmit.

Lower curve: Single-threshold VFR scheme
Upper curve: Double-threshold VFR scheme with the sum of the two thresholds G1 and G2 kept constant at 9 dB; G1 is plotted along the X-axis in this case.

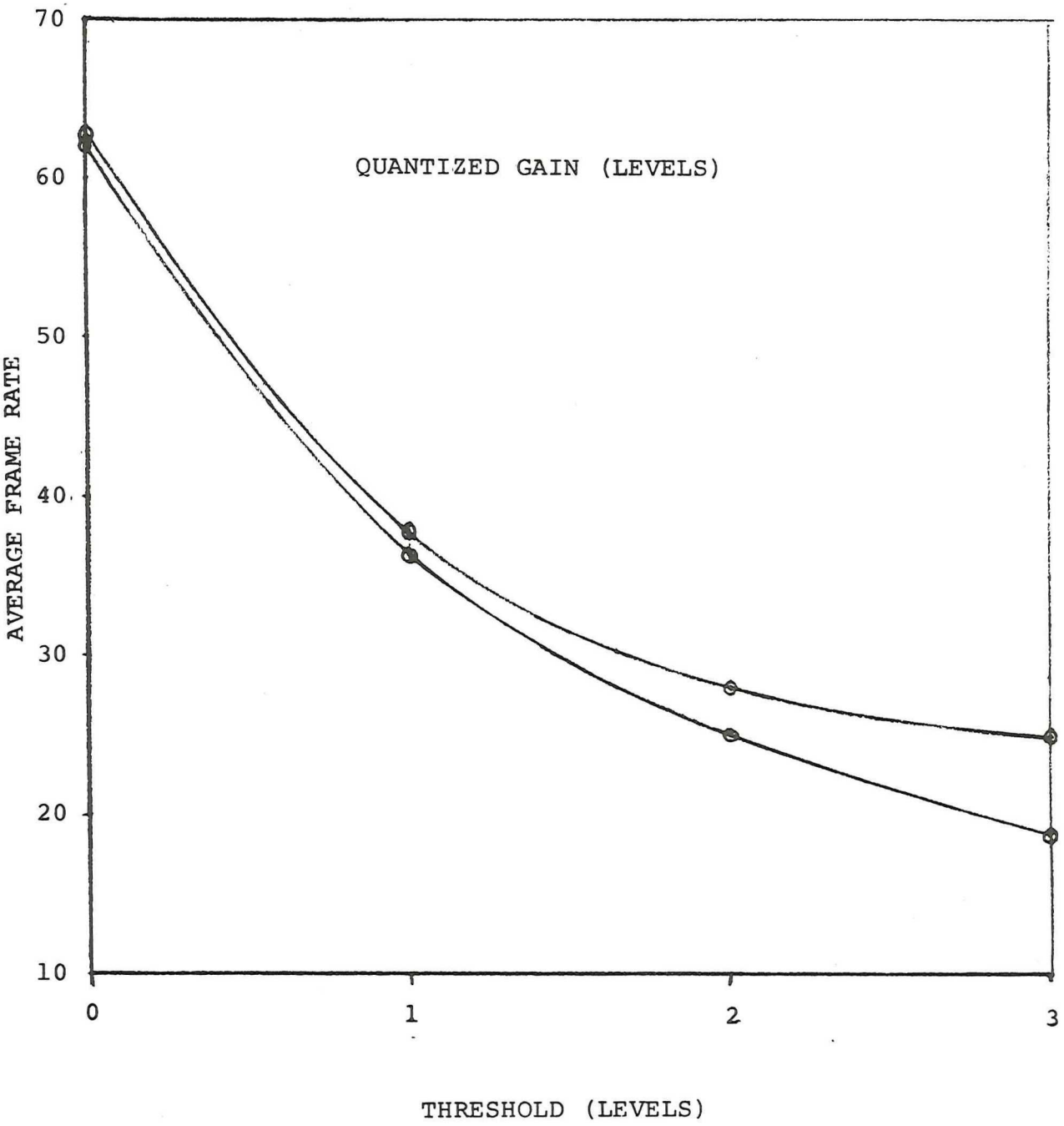

Fig. 5. Average frame rate of gain transmission as a function of threshold value employed in the VFR scheme which considers change in quantized gain level IG for deciding when to transmit.

Lower curve: Single-threshold VFR scheme
Upper curve: Double-threshold VFR scheme with the sum of the two thresholds IG1 and IG2 kept constant at 7; IG1 is plotted along the X-axis in this case.

rate savings are depicted in Fig. 6. (We used 5 bits for gain quantization.)

Results of Synthesis Experiments

We synthesized speech under all the six possible conditions in which each of the three sets of parameters, pitch, gain and log area ratios, was transmitted at either a fixed rate (100 fps) or at a variable frame rate. For log area ratios transmission, we used a fixed log likelihood ratio threshold of 1.5 dB, while we considered different thresholds for pitch and gain VFR schemes. We used informal listening tests for judging relative speech quality of synthesized speech.

We found that for a given frame rate of transmission, use of either the VFR scheme operating on unquantized parameter (pitch or gain) values or the VFR scheme operating on quantized levels produced about the same spech quality.

Comparing the fixed 100 fps transmission of pitch with the VFR transmission of the quantized pitch levels using a double-threshold scheme with thresholds given by 0 and 1, we actually perceived a slight quality improvement in the latter case over the former, even though the VFR transmission yielded an average frame rate of 35 fps only. The reason for this is that the fixed rate transmission often produces sequences of equal pitch values, while this does not happen in the VFR transmission since pitch is

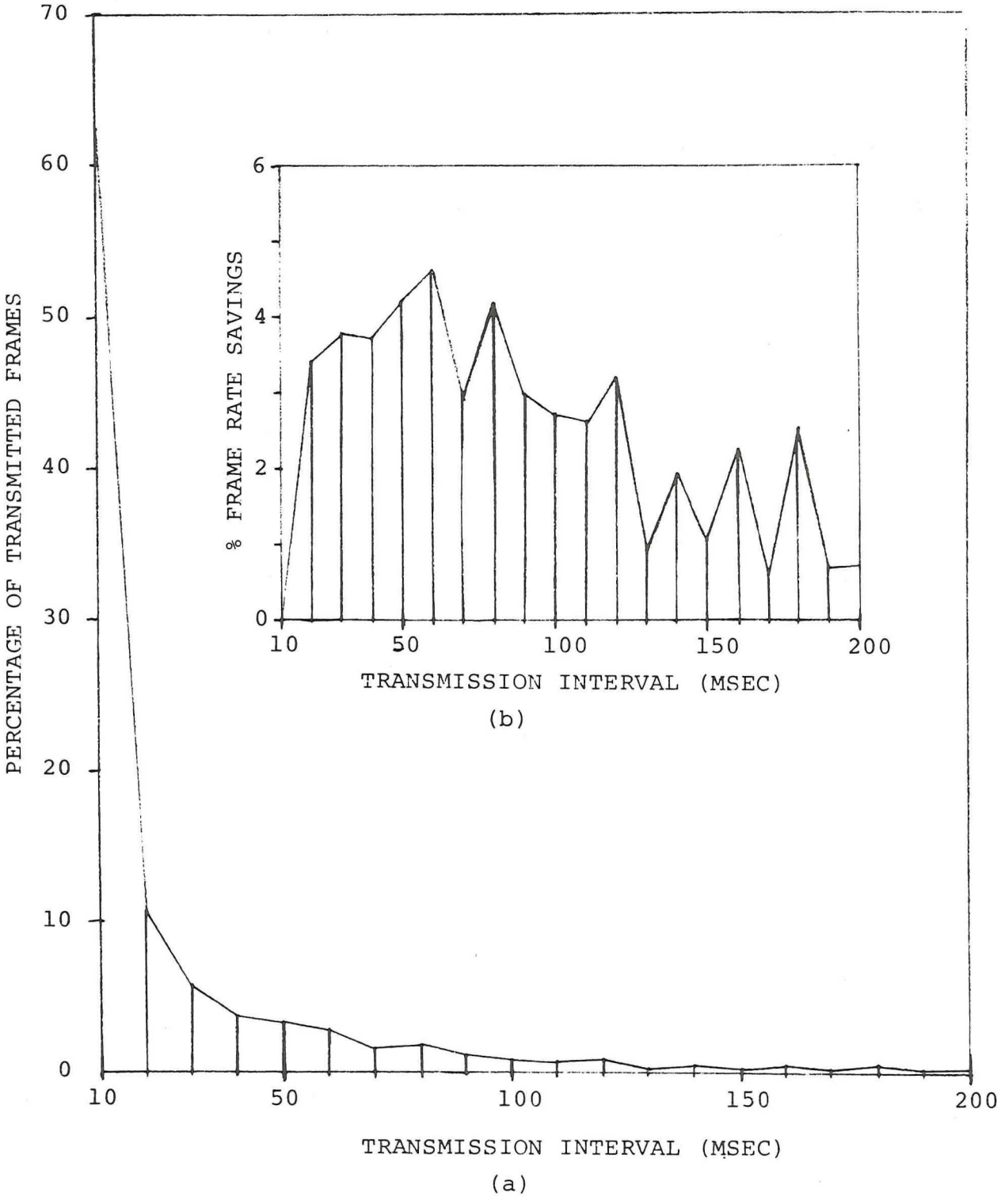

Fig. 6. Double-threshold VFR scheme for quantized gain levels with the two thresholds held at IG1=2, IG2=3.

Plot (a): Histogram of transmission interval

Plot (b): % Frame rate savings versus transmission interval

interpolated (rather than being repeated) at the receiver between transmissions. (Of course, no interpolation is done when adjacent transmitted frames are not both voiced; we simply repeat the last transmitted pitch in this case.)

Our main purpose of conducting synthesis experiments was to arrive at specific recommendations regarding the values of the thresholds employed in the VFR schemes for pitch and gain. Our criterion for selecting the threshold values was to reduce the frame rate of parameter transmission as much as possible without causing any noticeable change in speech quality relative to the fixed 100 fps transmission system (as judged by informal listening tests). Our choice of threshold values based on this subjective criterion are given in the next section.

IV. SPECIFIC RECOMMENDATIONS

A. Pitch

We recommend a double-threshold VFR scheme that operates on quantized levels, with the two thresholds given by IP1=0 and IP2=1. For the case when pitch is quantized using 6 bits, the average frame rate for pitch with this scheme is estimated to be about 35 fps. This average was computed over the 11-sentence data base we considered. Average over individual sentences varied between 26 fps and 50 fps.

B. Gain

For gain, we also recommend a double-threshold VFR scheme that operates on quantized levels, with the two thresholds given by IG1=2 and IG2=3. For the case when gain is quantized using 5 bits, the average frame rate for gain with this scheme is estimated to be about 32 fps. This average was computed over the 11-sentence data base we considered. Average over individual sentences varied between 12 fps and 45 fps.

C. Estimated Bit Rate for LPC-II

With 6 bits for quantizing pitch, 5 bits for gain, 36 bits for log area ratios, and 3 bits for header, and assuming average frame rates of 35, 32, 37 and 102 fps respectively for those four items, we estimate the average

bit rate of LPC-II vocoder to be about 2000 bps for continuous speech.

Reference

1. R. Viswanathan and J. Makhoul, "Specifications for ARPA-LPC System II," NSC Note 82, Feb. 1976.